

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/351842846>

# A Technique of Code Clone Detection based on Defined Mechanism for Threshold Calculation | A Technique of Code Clone Detection based on Defined Mechanism for Threshold Calculation

Article · May 2021

CITATIONS

0

READS

27

3 authors:



Rubab Mehboob

University of Engineering and Technology, Taxila

9 PUBLICATIONS 40 CITATIONS

SEE PROFILE



Sidra Shabbir

University of Engineering and Technology, Taxila

6 PUBLICATIONS 29 CITATIONS

SEE PROFILE



Ali Javed

University of Engineering and Technology, Taxila

109 PUBLICATIONS 953 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Square texton [View project](#)



Multimedia Forensics [View project](#)



# The Nucleus

A Quarterly International Scientific Journal

[HOME](#) [ABOUT](#) [LOGIN](#) [REGISTER](#) [SEARCH](#) [CURRENT](#)  
[ARCHIVES](#)

[Home](#) > [Vol 54, No 4 \(2017\)](#) > [Mehboob](#)

## A Technique of Code Clone Detection based on Defined Mechanism for Threshold Calculation

*R. Mehboob, S. Shabbir, A. Javed*

### Abstract

Over the past few years the revolution in the technology and use of programming languages for product development has made code reusability a common practice. Consequently the problem of code cloning is also increasing leading to redundancy and increased maintenance cost. The real motivation of the proposed research work is to identify code clones from pair of codes that are going to be utilized for a project under consideration. Existing practices such as control flow graphs (CFGs) and abstract syntax tree (AST) promote a high level of abstraction by masking the inner details of the code. Therefore, a strategy is needed to define the mechanism for calculation of the threshold value to identify clones by considering the inner details of the code. This paper presents a defined mechanism for the computation of threshold for code clone detection. Moreover, the inner details of the code are examined by performing the comparative analysis of tokens and conditional clauses. The proposed technique eliminates high level of abstraction caused by the use of CFGs. The proposed method is tested on custom dataset of different sorting algorithms. Experimental results indicate the effectiveness of the proposed method for code clone detection.

### Full Text:

[PDF](#)

### References

- P. Pradhan, A. K. Dwivedi and S. K. Rath, "Detection of design pattern using graph isomorphism and normalized cross correlation", 8th Int. Conf. Contemp. Comput., pp. 208213, 2015.
- W. Li, D. Li, C. Qiu and J. Hou, "Efficient metric vector-based code clone detection using function-calling tree", Int. J. Hybrid Inf. Technol., vol. 8, no. 11, pp. 139150, 2015.
- S. Gupta and P. C. Gupta, "A novel approach to detect duplicate code blocks to reduce maintenance effort", Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 4, pp. 311314, 2016.
- W. Li, H. Saidi, H. Sanchez and M. Sch, "Detecting similar programs via the Weisfeiler-Leman graph kernel", Proc. of 15th Int. Conf. on Software Reuse, pp. 315330, 2016.
- R. Tekchandani, R. Bhatia and M. Singh, "Semantic code clone detection for Internet of things applications using reaching definition and liveness analysis", J. Supercomput., pp. 128, 2016.
- R. Koschke, "Large-scale inter-system clone detection using suffix trees and hashing", J. Softw. Evol. Process, vol. 26, no. 8, pp. 747769, 2014.

[OPEN JOURNAL SYSTEMS](#)

[Journal Help](#)

#### USER

Username

Password

Remember me

#### NOTIFICATIONS

- [View](#)
- [Subscribe](#)

#### JOURNAL CONTENT

Search

Search Scope

All

#### Browse

- [By Issue](#)
- [By Author](#)
- [By Title](#)

#### FONT SIZE

#### INFORMATION

- [For Readers](#)
- [For Authors](#)
- [For Librarians](#)

A. Sheneamer and J. Kalita, "Code clone detection using coarse and fine-grained hybrid approaches", IEEE 7th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2015, pp. 472480, 2016.

I. Keivanloo, F. Zhang, and Y. Zou, Threshold-free code clone detection for a large-scale heterogeneous Java repository, IEEE 22nd Int. Conf. Softw. Anal. Evol. Reengineering, Proc. SANER 2015, pp. 201210, 2015.

A. Ashish, "Clones clustering using K-means", 10th Int. Conf. Intell. Syst. Control, pp. 16, 2016.

Y. Higo and S. Kusumoto, "How often do unintended inconsistencies happen? Deriving modification patterns and detecting overlooked code fragments", IEEE Int. Conf. Softw. Maintenance, pp. 222231, 2012.

K. Kanagalakshmi and R. Suguna, "Software refactoring technique for code clone detection of static and dynamic website", Int. J. Comp. Applications", vol. 107, no. 12, pp. 110, 2014.

S. Singh and S. Kaur, "A systematic literature review: Refactoring for disclosing code smells in object oriented software", Ain Shams Eng. J., 2016.

T. Kamiya, "An Execution-Semantic and Content-and-Context- Based Code-Clone Detection and Analysis", Software Clones (IWSC), IEEE 9th Int. Workshop, pp. 17, 2015.

K. E. Rajakumari and T. Jebarajan, "A novel approach to effective detection and analysis of code clones", Third Int. Conf. Innov. Comput. Technol., pp. 287290, 2013.

I. D. Baxter, A. Yahin, L. Moura, M. SantAnna and L. Bier, "Clone detection using abstract syntax trees", Proc. Int. Conf. Softw. Maint. (Cat. No. 98CB36272), pp. 368377, 1998.

C. Liu, C. Chen, J. Han and P. S. Yu, "GPLAG: detection of software plagiarism by program dependence graph analysis", Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. data Min., pp. 872881, 2006.

R. M. Abdel-Aziz, A. E. Aboutabl and M. S. Mostafa, "Clone detection using DIFF algorithm for aspect mining", Int. J. Adv. Comput. Sci. Appl., vol. 3, no. 8, pp. 137140, 2012.

W. Qu, Y. Jia and M. Jiang, "Pattern mining of cloned codes in software systems", Inf. Sci. (Ny), vol. 259, pp. 544554, 2014.

F. H. Su, J. Bell and G. Kaiser, "Challenges in behavioral code clone detection", IEEE 23rd Int. Conf. Softw. Anal. Evol. Reengineering, SANER 2016, vol. 2, pp. 2122, 2016.

A. Okutan and O. Taner Yildiz, "A novel kernel to predict software defectiveness", J. Syst. Softw., vol. 119, pp. 109121, 2016.

Z. Tian, T. Liu, Q. Zheng, M. Fan, E. Zhuang and Z. Yang, "Exploiting thread-related system calls for plagiarism detection of multithreaded programs", J. Syst. Softw., vol. 119, pp. 136148, 2016.

G. Maskeri, D. Karnam, S. A. Viswanathan and S. Padmanabhuni, "Version history based source code plagiarism detection in proprietary systems", IEEE Int. Conf. Softw. Maintenance, ICSM, pp. 609612, 2012.

E. Flores, A. Barron-Cedeno, L. Moreno and P. Rosso, "Cross-language source code re-use detection using latent semantic analysis", J. Univers. Comput. Sci., vol. 21, no. 13, pp. 17081725, 2015.

S. Stojanovi?, Z. Radivojevi?, and M. Cvetanovi?, "Approach for estimating similarity between procedures in differently compiled binaries", Inf. Softw. Technol., vol. 58, pp. 259271, 2015.

Y. Yuan, F. Zhang, and X. Su, "CloneAyz : An approach for clone representation and analysis", Inf. Sci. Control Engg. (IEEE), pp. 252-256, 2016.

G. Vale, E. Figueiredo, R. Abilio, and H. Costa, "Bad smells in software product lines: A systematic review", Proc. of 8th Brazilian Symp. Softw. Components, Archit. Reuse, pp. 8494, 2014.

A. Ouni, M. Kessentini, S. Bechikh and H. Sahraoui, "Prioritizing code-smells correction tasks using chemical reaction optimization," Software Quality Journal , vol. 23, no. 2. 2015.

A. Yamashita and L. Moonen, "Do code smells reflect important maintainability aspects?", IEEE Int. Conf. Softw. Maintenance, ICSM, pp. 306315, 2012.

F. Hermans, M. Pinzger and A. van Deursen, "Detecting and refactoring code smells in spreadsheet formulas", Empir. Softw. Eng., vol. 20, no. 2, pp. 549575, 2015.

B. Hauptmann, M. Junker, S. Eder, L. Heinemann, R. Vaas and

P. Braun, "Hunting for Smells in Natural Language Tests", Proc. of Int. Conf. on Software Engg., no. 1, pp. 47, 2013.

Z. Li, S. Lu, S. Myagmar and Y. Zhou, "CP-Miner: Finding copy-paste and related bugs in large-scale software code", IEEE Trans. Softw. Eng., vol. 32, no. 3, pp. 176192, 2006.

H. Kaur and R. Maini, Identification of recurring patterns of code to detect structural clones", Proc. of 6th Int. Adv. Comput. Conf.,

pp. 398403, 2016.

M. Abdelkader and M. Mimoun, "Clone detection using time series and dynamic time warping techniques", Third World Conf. Complex Syst., pp. 16, 2015.

J. Alnihoud and R. Mansi, "An enhancement of major sorting algorithms", Int. Arab J. Inf. Technol., vol. 7, no. 1, pp. 5562, 2010.

---

## Refbacs

- There are currently no rebfacs.